НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. Ігоря СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

Звіт з виконання лабораторної роботи

ДОСЛІДЖЕННЯ РЕАЛІЗАЦІЙ ПРОТОКОЛУ SSL

Виконали студентки
групи ФІ-32мн
Зацаренко А. Ю.
Футурська О.В.

Київ — 2024

ЗВІТ

1.1 Мета комп'ютерного практикуму

Дослідження особливостей реалізації криптографічних механізмів протоколу SSL/TLS.

1.2 Постановка задачі

Розробити програмний засіб захисту логічного каналу зв'язку, що використовує протокол TLS (версія 1.3). Програмний засіб повинен мати хмарну архітектуру, в якій клієнт повинен бути орієнтованим на операційні системи та платформи для мобільних телефонів та планшетних комп'ютерів. Дозволяється використання бібліотеки OpenSSL або Crypto++ — реалізації з відкритим кодом.

Клієнт має підтримувати ОС Android 11.0 або вище, або ОС iOS 17.0 або вище. Серверна частина має підтримувати ОС Windows (версія ядра не менша за 10). Програмний засіб має реалізовувати автентифікацію обох сторін згідно з протоколом TLS. Реалізація протоколу має детально відображати процес формування нового сеансу та відновлення існуючого (має відображатися кожен з типів пакетів, його вміст та всі параметри протоколу як на клієнті, так і на серверній частині), включаючи процес перевірки сертифікатів. Програмний засіб має надавати змогу користувачу безпосередньо впливати на цей процес з кожної із сторін, а саме блокувати перевірку сертифікатів, задавати конкретні значення випадкових змінних, задавати та згодом змінювати криптографічні методи, що використовуються. Після встановлення логічного каналу зв'язку програмний засіб повинен дозволяти проводити обмін довільними повідомлення за цим каналом. Коректність реалізації необхідно підтвердити за допомогою використання програми для перехоплення та аналізу пакетів — Wireshark. Також необхідно продемонструвати атаку sslstrip на власну реалізацію протоколу (можна використовувати допоміжні програми) та

обґрунтувати результати застосування.

1.3 Хід роботи

Протоколи SSL (Secure Sockets Layer) і TLS (Transport Layer Security) забезпечують безпечну передачу даних через мережу, використовуючи криптографічні механізми для шифрування, автентифікації та забезпечення цілісності даних.

1.3.1 Структура протоколів SSL/TLS та їх підпротоколів

Основна структура поділяється на два рівні:

1. Записний рівень (Record Layer) — це базовий рівень, який забезпечує:

• Шифрування даних: шифрує дані за допомогою симетричних алгоритмів (AES, ChaCha20).

• Компресію даних (у ранніх версіях SSL/TLS, відсутня у TLS 1.3).

• Забезпечення цілісності: використовує MAC (Message Authentication Code) для перевірки даних.

• Розбиття даних на пакети.

2. Протоколи високого рівня — ці підпротоколи працюють поверх записного рівня:

• Handshake Protocol:

– Використовується для встановлення нового сеансу.

– Забезпечує автентифікацію сторін (сертифікати X.509).

– Узгоджує параметри шифрування (ключі, алгоритми, версії протоколу).

• Change Cipher Spec Protocol (не використовується в TLS 1.3):

– Сигналізує про перехід до використання узгоджених параметрів шифрування.

• Alert Protocol:

– Відповідає за передачу попереджень або помилок (наприклад, невірний сертифікат).

– Може бути критичним (terminate connection) або некритичним.

• Application Data Protocol:

– Передає зашифровані дані прикладного рівня (HTTP, SMTP тощо).

Формування нового сеансу (Handshake)

Handshake Protocol у TLS 1.3 спрощений у порівнянні з попередніми версіями та має наступні етапи:

1) ClientHello (Клієнт вітається):

• Клієнт надсилає підтримувані алгоритми шифрування, версію протоколу, випадкову змінну (random).

2) ServerHello (Сервер вітається):

• Сервер відповідає вибраним алгоритмом, своєю версією протоколу та випадковою змінною.

3) Key Exchange (Обмін ключами):

• Сторони виконують обчислення загального секрету (shared secret) на основі алгоритму обміну ключами (ECDHE).

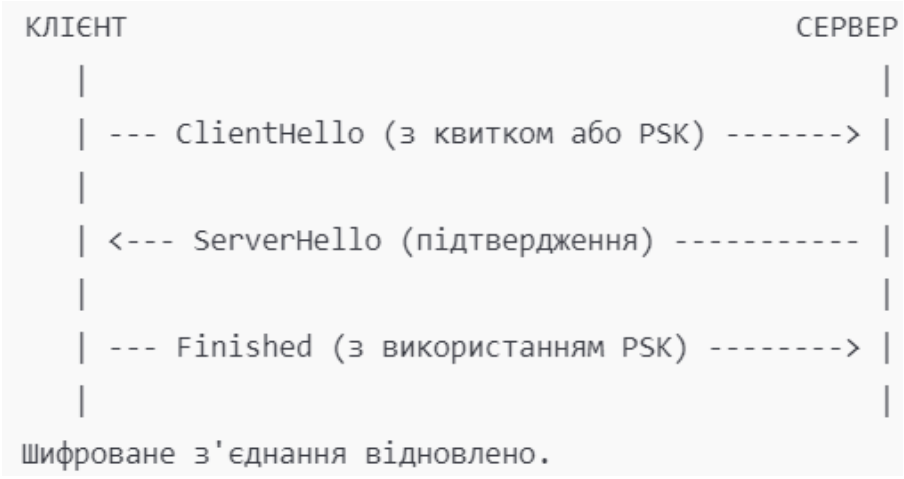
• Сервер надсилає свій сертифікат для автентифікації.

4) Finished (Завершення):

• Клієнт і сервер завершують обмін повідомленнями, перевіряючи цілісність обміну даних.

Відновлення існуючого сеансу (Session Resumption)

У TLS 1.3 існує два методи відновлення:



1) Session Tickets:

• Сервер надсилає клієнту квиток (ticket), що містить параметри попереднього сеансу.

• Клієнт використовує цей квиток для відновлення з'єднання.

2) Pre-shared Keys (PSK):

• Клієнт і сервер узгоджують попередньо збережений ключ (pre-shared key).

1.3.2 Порівняння версій протоколів від SSL 1.0 до TLS 1.3

Протоколи SSL і TLS еволюціонували з часом, щоб усунути вразливості попередніх версій, покращити продуктивність і забезпечити більшу безпеку. Розглянемо кожну версію:

1. SSL 1.0 (1995, не був офіційно випущений)

• Стан: Відхилений на стадії розробки через серйозні вразливості.

• Особливості:

* Ненадійний захист шифрування.

* Погана взаємодія між шифруванням і передачею даних.

- Причина невикористання:
  * Розробники Netscape виявили критичні помилки безпеки.

2. SSL 2.0 (1995)

- Стан: Застарілий (використання заборонено).
- Особливості:
  * Підтримка обмеженої кількості алгоритмів.
  * Дані передавалися у відкритому вигляді перед встановленням шифрування.
  * Відсутність захисту від перехоплення сесій.
- Вразливості:
  * Відсутність перевірки цілісності даних.
  * Атаки на повернення до менш безпечних алгоритмів.

3. SSL 3.0 (1996)

- Стан: Застарілий (заміщений TLS 1.0).
- Покращення:
  * Впровадження HMAC для перевірки цілісності даних.
  * Шифрування повідомлень перед передачею.
- Вразливості:
  * Атака POODLE (Padding Oracle On Downgraded Legacy Encryption).
  * Відсутність захисту від атак даунгрейду.

4. TLS 1.0 (1999)

- Стан: Застарілий (відключений у 2020 році більшістю систем).
- Відмінності від SSL 3.0:
  * Покращена структура записного рівня.
  * Впровадження шифрування на основі ключа (Key Material Derivation).
  * Захист від атак на цілісність (MAC використовується після шифрування).
- Вразливості:
  * Атака BEAST (Browser Exploit Against SSL/TLS).

5. TLS 1.1 (2006)

- Стан: Застарілий (рекомендовано не використовувати).
- Покращення:
  * Введення IV (Initialization Vector) для захисту від повторів.
  * Кращий захист проти атак BEAST.
- Вразливості:
  * Вразливість до атак на певні алгоритми шифрування (наприклад, RC4).

6. TLS 1.2 (2008)

- Стан: Широко використовується (станом на 2024 рік).
- Покращення:
  * Підтримка нових алгоритмів шифрування (AES, GCM).
  * Захист від атак на MAC.
  * Підтримка SHA-256 для перевірки хешів.
- Особливості:
  * Використовується більшістю вебсайтів і систем.
- Вразливості:
  * Вразливість до певних типів атак (BREACH, Lucky 13).

7. TLS 1.3 (2018)

- Стан: Рекомендований до використання (найсучасніша версія).
- Основні переваги:
  * Скорочення Handshake: зменшена кількість раундів до одного.
  * Виключені небезпечні алгоритми (RC4, MD5, SHA-1).
  * Підтримка сучасних криптографічних алгоритмів (ChaCha20, AES-GCM).
  * Захист від атак на шифрування (Forward Secrecy за замовчуванням).
  * Виключення застарілих функцій (Change Cipher Spec).
- Недоліки:
  * Не підтримує сумісність зі старими системами.

Таблиця 1.1 – Порівняння протоколів SSL/TLS

Характеристика	SSL 2.0	SSL 3.0	TLS 1.0	TLS 1.1	TLS 1.2	TLS 1.3
Рік випуску	1995	1996	1999	2006	2008	2018
Захист цілісності	Ні	Так	Так	Так	Так	Так
Алгоритми шифрування	Обмежені	RC4, DES	AES, RC4	AES	AES, GCM	AES, ChaCha20
Захист від атак даунгрейду	Ні	Ні	Частково	Так	Так	Так
Скорочений Handshake	Ні	Ні	Ні	Ні	Ні	Так
Підтримка Forward Secrecy	Ні	Ні	Частково	Частково	Так	Так

1.3.3 Структуру сертифіката, ролі кореневих сертифікатів, акредитованих центрів сертифікації ключів (CA), списків CRL та протоколу OSCP

Сертифікат — це цифровий документ, який підтверджує автентичність публічного ключа та ідентичність суб'єкта, що його використовує. Сертифікати зазвичай відповідають стандарту X.509 і мають наступну структуру:

1) Версія — визначає стандарт сертифіката (наприклад, X.509 версії 1, 2 або 3). У сучасних сертифікатах використовується X.509 версія 3.

2) Серійний номер — унікальний ідентифікатор сертифіката, наданий центром сертифікації (CA).

3) Алгоритм підпису — алгоритм, яким CA підписує сертифікат (наприклад, RSA, ECDSA з SHA-256).

4) Ім'я видавця (Issuer) — інформація про центр сертифікації, який видав сертифікат.

5) Період дії (Validity) — часовий проміжок, протягом якого сертифікат вважається дійсним:

- Not Before: дата початку дії.
- Not After: дата закінчення дії.

6) Ім'я суб'єкта (Subject) — інформація про власника сертифіката (ім'я,

доменне ім'я, адреса).

7) Публічний ключ (Public Key Information) — включає сам публічний ключ і алгоритм, який використовується для шифрування (наприклад, RSA, ECC).

8) Розширення (Extensions) — у X.509 версії 3 додаються розширення, такі як:

- Key Usage: дозволені функції ключа (підпис, шифрування, автентифікація).
- Subject Alternative Name (SAN): додаткові імена доменів або IP-адреси.
- Basic Constraints: визначає, чи є сертифікат CA.

9) Цифровий підпис (Signature) — підпис, виконаний CA, що підтверджує цілісність сертифіката.

Ролі в інфраструктурі сертифікації

Кореневі сертифікати (Root Certificates):

- Видаються авторитетним центром сертифікації (CA) і служать базою для довіри.
- Зазвичай попередньо вбудовані в операційні системи, браузери та інші платформи.
- Високий рівень довіри — якщо кореневий сертифікат скомпрометовано, довіра до всіх його похідних сертифікатів втрачається.

Центри сертифікації (Certificate Authorities, CA):

- Організації, які видають цифрові сертифікати.
- Види:
  * Кореневі CA: створюють самопідписані сертифікати.
  * Проміжні CA (Intermediate CA): використовуються для розподілу навантаження і додаткової безпеки (підписуються кореневим CA).
- Завдання:
  * Перевірка ідентичності суб'єктів.
  * Видача та відкликання сертифікатів.

Списки відкликаних сертифікатів (CRL, Certificate Revocation List):

- Списки сертифікатів, які були відкликані до завершення їх терміну дії.
- Причини відкликання:
  * Компрометація ключа.
  * Неправильна інформація в сертифікаті.
  * Припинення діяльності суб'єкта.
- Основні недоліки:
  * Затримка в оновленні.
  * Потреба в завантаженні великих списків.

Протокол перевірки статусу сертифіката (OCSP, Online Certificate Status Protocol):

- Забезпечує перевірку статусу сертифіката в реальному часі.
- Клієнт надсилає запит до OCSP-сервера, який повідомляє, чи є сертифікат дійсним, відкликаним або невідомим.
- Переваги:
  * Швидкість перевірки.
  * Зменшення навантаження в порівнянні з CRL.
- Недоліки:
  * Якщо OCSP-сервер недоступний, можуть виникати затримки у з'єднанні.

Ієрархія сертифікатів

1. Кореневий CA: самопідписаний сертифікат.

2. Проміжний CA: підписується кореневим CA, забезпечує додаткову безпеку.

3. Кінцевий сертифікат: використовується клієнтами (наприклад, сертифікати для вебсайтів).

1.3.4 Криптографічні методи, що використовуються в SSL і TLS

Протоколи SSL та TLS використовують комбінацію криптографічних методів для забезпечення захищеної передачі даних через незахищені канали. Розглянемо основні функції та алгоритми:

1. Узгодження ключа (Key Exchange)

Ця фаза забезпечує безпечне встановлення спільного сеансового ключа між клієнтом і сервером.

Методи:

1) RSA (Rivest-Shamir-Adleman):

- Використовується для шифрування сеансового ключа.
- Недоліки: вразливість до атак на основу криптографії (факторизація) і відсутність Forward Secrecy.

2) DH (Diffie-Hellman):

- Базується на обчислювальній складності проблеми дискретного логарифму.
- Переваги: забезпечує Forward Secrecy.
- Недоліки: статичний DH вразливий до атак при використанні слабких параметрів.

3) ECDH (Elliptic Curve Diffie-Hellman):

- Покращена версія DH, яка базується на еліптичних кривих.
- Менший розмір ключів, але високий рівень безпеки.
- Переваги: ефективність і Forward Secrecy.

4) PSK (Pre-Shared Key):

- Використовує заздалегідь встановлений спільний ключ.
- Недоліки: складність управління ключами.

5) DHE (Diffie-Hellman Ephemeral):

- Варіант DH із одноразовими ключами.
- Переваги: Forward Secrecy.

6) ECDHE (Elliptic Curve Diffie-Hellman Ephemeral):

• Ефективний варіант DH з еліптичними кривими та одноразовими ключами.

• Рекомендовано для TLS 1.2 та TLS 1.3.

2. Автентифікація (Authentication)

Цей процес перевіряє ідентичність сторін за допомогою цифрових сертифікатів. Методи:

1) Цифрові сертифікати X.509:

• Використовуються для ідентифікації сервера і, за необхідності, клієнта.

• Засновані на ієрархії довіри (CA, кореневі сертифікати).

2) RSA:

• Використовується для автентифікації сервера.

• Сервер підписує дані відкритим ключем.

3) DSA (Digital Signature Algorithm):

• Алгоритм для створення цифрових підписів.

4) ECDSA (Elliptic Curve Digital Signature Algorithm):

• Ефективна версія DSA з використанням еліптичних кривих.

3. Шифрування (Encryption)

Методи:

1) Блочні шифри:

• DES (Data Encryption Standard): Вразливий через малу довжину ключа (56 біт).

• 3DES (Triple DES): Покращений DES, але має низьку швидкість.

• AES (Advanced Encryption Standard): Широко використовується (128, 192, 256 біт).

• Camellia: Альтернатива AES з подібними властивостями.

2) Потокові шифри:

• RC4: Раніше використовувався, але визнаний небезпечним через вразливість до атак.

3) Сучасні шифри (TLS 1.3):

• AES-GCM: Автентифіковане шифрування (шифрування та

MAC).

• ChaCha20-Poly1305: Ефективний для мобільних пристроїв і забезпечує високу швидкість.

4. Контроль цілісності (MAC, Message Authentication Code)

MAC гарантує, що дані не були змінені під час передачі. Методи:

1) HMAC (Hash-Based Message Authentication Code):

• Використовує хеш-функції (SHA-1, SHA-256, SHA-384).

• Популярний у всіх версіях TLS.

2) CBC-MAC (Cipher Block Chaining MAC):

• Використовувався в SSL/TLS разом із блочними шифрами.

3) GMAC (Galois Message Authentication Code):

• Використовується разом із AES-GCM у TLS 1.2 і TLS 1.3.

Таблиця 1.2 – Еволюція криптографічних методів у версіях TLS

Метод	SSL 2.0	SSL 3.0	TLS 1.0	TLS 1.1	TLS 1.2	TLS 1.3
Узгодження ключа	RSA, DH	RSA, DH	RSA, DHE	RSA, DHE, ECDHE	RSA, DHE, ECDHE	ECDHE, PSK
Аутентифікація	RSA	RSA, DSA	RSA, DSA	RSA, DSA, ECDSA	RSA, ECDSA	ECDSA, PSK
Шифрування	RC4, DES	3DES, RC4, AES	3DES, AES	AES	AES-GCM, Camellia	AES-GCM, ChaCha20
MAC	MD5	SHA-1, MD5	SHA-1	SHA-1, SHA-256	SHA-256, SHA-384	Вбудовано в AEAD

1.3.5 Опис і класифікація вразливостей протоколів SSL і TLS

Протоколи SSL/TLS мають низку відомих вразливостей, які можна класифікувати за типами атак: на криптографію, логіку протоколу, реалізацію та налаштування.

1. Атаки на логіку протоколу

Атака переузгодження (Renegotiation Attack)

– Суть: Протокол дозволяє сторонам переузгоджувати параметри з'єднання. Атака дозволяє зловмиснику вставляти свої дані на початку легітимної сесії.

– Результат: Можливе виконання атак "людина посередині"(MITM) або ін'єкція даних.

– Захист: Заборона переузгодження або перевірка контексту сесії.

Атака відкату версії (Version Rollback Attack)

– Суть: Зловмисник змушує клієнта і сервер використовувати застарілу (уразливу) версію SSL/TLS.

– Результат: Можливе використання слабких алгоритмів.

– Захист: Механізм перевірки версії (з TLS 1.3 видалено підтримку старих версій).

2. Атаки на криптографію

Атака на шифр RC4

– Суть: RC4 має слабке початкове налаштування, що дозволяє зловмиснику відновити текст із шифротексту після аналізу великої кількості даних.

– Результат: Розшифрування переданих даних.

– Захист: Відмова від використання RC4 на користь AES-GCM або ChaCha20.

Атака на генератор псевдовипадкових чисел (PRNG)

– Суть: Недосконалий PRNG може призвести до передбачуваності сесійних ключів.

– Результат: Компрометація конфіденційності сесії.

– Захист: Використання якісних PRNG (наприклад, NIST SP800-90A DRBG).

3. Атаки на налаштування

Атака sslstrip

– Суть: Зловмисник змінює HTTPS-з'єднання на HTTP, перехоплюючи дані без шифрування.

– Результат: Викрадення чутливих даних.

– Захист: Використання HSTS (HTTP Strict Transport Security).

Атака BEAST (Browser Exploit Against SSL/TLS)

– Суть: Уразливість у режимі CBC (Cipher Block Chaining), що дозволяє зловмиснику відновити дані зі шифротексту.

– Результат: Розшифрування конфіденційної інформації.

– Захист: Використання AES-GCM (або TLS 1.1 і вище).

Атака CRIME (Compression Ratio Info-leak Made Easy)

– Суть: Використання стиснення даних для виявлення зашифрованих частин повідомлення через аналіз розміру.

– Результат: Розкриття чутливих даних (наприклад, cookies).

– Захист: Вимкнення TLS-компресії.

Атака BREACH (Browser Reconnaissance and Exfiltration via Adaptive Compression of Hypertext)

– Суть: Атака на HTTP-компресію аналогічна до CRIME.

– Результат: Викрадення даних із HTTP-заголовків.

– Захист: Додавання випадкових даних у відповіді сервера (padding).

Атака POODLE (Padding Oracle On Downgraded Legacy Encryption)

– Суть: Використання уразливості в алгоритмі паддінгу SSL 3.0 для дешифрування даних.

– Результат: Викрадення даних.

– Захист: Відключення SSL 3.0.

Lucky 13

– Суть: Атака на таймінги в режимі CBC, що дозволяє отримати доступ до зашифрованих даних.

– Результат: Розшифрування частин повідомлення.

– Захист: Використання AEAD (наприклад, AES-GCM).

4. Атаки на даунгрейд

FREAK (Factoring RSA Export Keys)

– Суть: Зловмисник змушує сторони використовувати слабкі 512-бітні ключі, доступні через політику експортних шифрів.

– Результат: Розшифрування даних.

– Захист: Відмова від підтримки експортних шифрів.

Logjam

– Суть: Використання слабких параметрів Diffie-Hellman для перехоплення з'єднання.

– Результат: Розшифрування трафіку.

– Захист: Використання 2048-бітних і більших параметрів DH.

5. Атаки на реалізацію

Heartbleed

– Суть: Уразливість у реалізації OpenSSL (функція Heartbeat), що дозволяє отримати доступ до пам'яті сервера.

– Результат: Викрадення ключів, сертифікатів та інших конфіденційних даних.

– Захист: Оновлення OpenSSL.

BERserk

– Суть: Помилка у перевірці підпису в бібліотеці Mozilla NSS.

– Результат: Зловмисник може створити підроблений підпис, що виглядає дійсним.

– Захист: Оновлення бібліотек.

Таблиця 1.3 – Класифікація відомих вразливостей SSL/TLS

Атака	**Тип атаки**	**Результат**	**Захист**
Renegotiation	Логіка протоколу	Ін'єкція даних, MITM	Заборона переузгодження
Version Rollback	Логіка протоколу	Використання слабких алгоритмів	TLS 1.3
RC4	Криптографія	Розшифрування даних	AES-GCM, ChaCha20
PRNG	Криптографія	Передбачуваність ключів	Якісні PRNG
sslstrip	Налаштування	Перехоплення HTTP-трафіку	HSTS
BEAST	Криптографія	Розшифрування повідомлень	AES-GCM
CRIME	Стиснення	Розкриття даних	Вимкнення TLS-компресії
BREACH	Стиснення	Викрадення HTTP-заголовків	Padding
POODLE	Логіка протоколу	Викрадення даних	Відключення SSL 3.0
Lucky 13	Таймінги	Розшифрування даних	AEAD
FREAK	Даунгрейд	Розшифрування даних	Відключення експортних шифрів
Logjam	Даунгрейд	Викрадення даних	Сильні параметри DH
Heartbleed	Реалізація	Викрадення пам'яті сервера	Оновлення OpenSSL
BERserk	Реалізація	Підробка підпису	Оновлення бібліотек

ВИСНОВКИ

У ході виконання лабораторної роботи було досягнуто наступних результатів:

1) Досліджено протоколи SSL/TLS. Вивчено структуру протоколів, їх підпротоколи, функції, а також процеси встановлення та відновлення сеансів зв'язку.

2) Порівняно версії протоколів. Проведено порівняльний аналіз версій SSL 1.0–3.0 та TLS 1.0–1.3. Визначено переваги сучасних реалізацій та їх внесок у підвищення безпеки.

3) Виконано огляд вразливостей. Досліджено основні відомі вразливості протоколів SSL/TLS, зокрема атаки на криптографію, логіку протоколу, реалізацію та налаштування. Запропоновано методи захисту, які можуть бути застосовані для зниження ризиків.

Аналіз протоколу TLS 1.3 дозволили глибше зрозуміти принципи функціонування захищених комунікацій. Сучасні протоколи забезпечують високий рівень безпеки, проте залишаються вразливими до помилок реалізації, налаштувань або специфічних атак. Отримані знання можуть бути використані для покращення безпеки інформаційних систем.